2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Generating Reactive Approach Motions Towards Allowable Manifolds using Generalized Trajectories from Demonstrations

Cristian Vergara[1,2], Santiago Iregui[1,2], Joris De Schutter[1,2] and Erwin Aertbeliën[1,2]

***Abstract*— There is a high cost associated to the time and expertise required to program complex robot applications with high variability. This is one of the main barriers that inhibit the entry of robotic automation in small and medium-sized enterprises. To tackle the high level of task uncertainty associated with changing conditions of the environment, we propose a framework that leverages a combination between learning from demonstration (LfD) and constraint-based task specification and control. This synergy enables our framework to use LfD to generalize reactive approach motions (RAMo) towards not only a single pose but towards an allowable manifold defined with respect to the object to interact with. As a result, the robot executes the task by following a feasible approach motion generalized from the learned information. This approach motion is generated based on an initial representation of the environment, and it can be reactively adapted in function of current updates of the environment using sensor information. The proposed framework enables the system to deal with applications that involve a high level of uncertainty, increasing the flexibility and robustness, compared to traditional sense-plan-act paradigms.**

I. Introduction

One of the main challenges of implementing robots in real-world applications is the need for intelligent algorithms that interpret multi-sensor information, thereby enabling robot systems to adapt to dynamic environments. These systems should exhibit the necessary predictability, robustness, and flexibility to deal with geometric uncertainty associated with the robot, task, and environment.

In view of increasing system predictability, in [1], we introduced a combination of the constraint-based methodology *expression-graph based Task Specification Language* (eTaSL) [2] and a Learning from Demonstration approach based on *Probabilistic Principal Component Analysis* (PPCA) [3]. This synergy enables us to use PPCA to generate trajectories defined by a linear combination of learned basis functions weighted by a set of *Degrees Of Freedom* (DOF), while leveraging the definition of eTaSL feature variables to specify these and other DOF associated with the task.

In this paper, we present *Reactive Approach Motions* (RAMo) that further extend the trajectory generation in [1] from specifying a static target pose toward a dynamic target pose defined on a modeled manifold that can be bounded. A robot programmer can intuitively define this manifold based on inherent task information such as symmetries related to the tool and the work piece. In addition, we increase the flexibility against dynamic obstacles in the environment by enabling local deformations of trajectory segments for instance to enable a tool to circumvent obstacles in different ways.

Fig. 1. Snapshots of a human–robot collaborative bin picking application. a) A robot is commanded to align with an augmented surface of the red sphere. During the approach motion, the generated trajectory is reactively adapted to avoid collisions with the bin and the clutter. b) A human is able to move the end effector by interacting with proximity sensors. Collisions with the wall and the clutter are continuously avoided while maintaining alignment with the augmented surface of the red sphere.

Hence, as the main contribution of this paper, we extend [1] towards the generation of RAMo, tightly integrating:

1) generation of reactively adaptable trajectories that preserve information provided by human demonstrations while still enabling deformations defined by controlled DOF that affect trajectory segments in their local neighborhood.
2) reactive adaptation of the trajectory goal pose within an allowable manifold defined in function of DOF that can be intuitively specified, bounded and constrained to preferable values. This manifold is specified using a modular approach where geometric features of the tool, the work piece, and the trajectory are specified separately, while enabling their automatic composition.

The proposed approach increases the flexibility and robustness, enabling the system to deal with applications that involve a high level of uncertainty associated with avoidance of dynamic obstacles.

The framework was evaluated in two use cases that involved the generation and control of RAMo to grasp objects onto which manifolds were defined for the allowable target grasp poses: a simulation use case with dynamic obstacles; and an experiment involving human-robot collaborative bin-

All authors gratefully acknowledge the financial support by the European Community's Seventh Framework Programme project Factory-in-a-day (FP7-609206), Flanders Make Projects YVES and SmartHandler_ICON, and KU Leuven C1 project "Neurophysiological investigations of the human and nonhuman primate brain: from perception to action" (3M180301). *(Corresponding author: Cristian Alejandro Vergara Perico)* cristian.vergara@kuleuven.be

[1] Robotics Research Group, KU Leuven. [2] Core Lab ROB, Flanders Make@KU Leuven

picking in which the environment was sensed by an industrial sensor while the human was perceived with proximity sensors as in [4] (see Fig 1).

II. Related Work

Representations to describe geometrical relations between rigid bodies have been addressed in literature since the early days of robotic research. In [5] the authors proposed mathematical derivations to describe spatial relations between rigid bodies in a goal state. The aforementioned approach inspired the instantaneous Task Specification using Constraints (iTaSC) [6] which extends the Task Frame Formalism [7] towards the definition of geometric body relations as a function of feature frames (virtual frames). This approach enables a systematic constraint-based task specification that deals with complex sensor-based robot behaviors in presence of uncertainties. In [2] the concept of *feature frames* was further extended towards the specification of the robot controller as a function of *feature variables*, which are used as auxiliary variables that enable the specification of DOF of the robot task. This paper exploits the concept of *feature variables* to intuitively define allowable goal-pose manifolds as parametric relationships between rigid bodies, thereby enabling the definition of RAMo that take into account sensor input as well as robot and environmental constraints.

In view of the search for feasible and stable grasping poses, several methodologies have leveraged the use of manifolds to represent grasping search regions. This is the case in [8], in which the authors used an exhaustive search for a unique stable grasping pose from a predefined set of possibilities. The lack of reactivity to adapt the grasping pose on-line typically limits the robustness of the system against uncertainties. To deal with this, more advanced systems have used a continuous search for the grasping pose. For instance, the *positioning mobile with respect to fixed frame* (PMF) method [9] introduced allowable manifolds that restrict, totally or partially, motion between two rigid bodies by specifying relations between geometric entities. This approach was further extended in [10] and [11] by also considering relations where the rotation and translation cannot be separated, as well as the composition with additional inequality constraints such as collision avoidance and/or joint limit constraints. Other approaches in literature have leveraged the use of available constraint-based frameworks. This is the case in [12], in which the authors proposed a systematic composition of constraints using eTaSL [2] that takes into account expressions between geometric entities, as well as the runtime monitoring of these constraints. Similarly, in [13] grasping envelopes were defined based on the specification of geometric constraints that are hierarchically solved using the constraint-based framework *stack-of-tasks* (SoT) [14]. Although these methods consider the definition of dynamic goal poses constrained to modeled manifolds, they do not address the generation of reactive approach trajectories that consider context information from the environment. Moreover, in our view, our approach simplifies the commissioning of the task by decomposing the manifold definition in DOF associated with the tool and with the workpiece.

Other methods focused more on the generalization of trajectories that can reactively adapt to changing conditions of the environment while preserving smooth dynamics. For instance, in [15], the authors developed a method to select the appropriate coupling terms of a dynamic movement primitive DMP [16] based on a neural network trained with synthetic collision-free trajectories and a minimal representation of the obstacles to dynamically avoid collision in different start-to-goal scenarios. This method nominally generates linear motions and adapts them relying only on a point cloud representation of the environment. Context information of the environment can be added by leveraging LfD, thereby alleviating problems associated with incorrect information from the vision system. For instance, in [17], authors shaped a reactive Lyapunov controller towards an attractor using demonstrations encoded by Gaussian Mixture Models (GMMs). Similarly, in [18], the authors proposed a reinforcement learning algorithm to improve upon demonstrated trajectories when solutions to new situations need to be found, such as for generating collision-free trajectories in dynamic environments. In contrast to these approaches, in this paper, we do not define a unique goal pose for the trajectory, but we specify a dynamic goal pose that is lying on an intuitively modeled manifold.

Few methodologies have addressed the trajectory generation towards goal poses defined on allowable manifolds. For instance, in [19], the authors proposed to generate trajectories that are not constrained to a unique goal pose, but to a *Task Space Region* [20]. In [21], the authors proposed a more reactive approach by leveraging Riemannian geometry to formulate dynamical systems. In contrast to the methods above, our approach leverages the intuitive definition of the task using DOF that can be monitored and controlled. As a result, our system obtains increased predictability and explainability while facilitating the task specification.

III. Reactively Adaptable Trajectories

In this section, we introduce adaptable trajectories $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})$ parameterized in function of DOF corresponding to the normalized path coordinate $0 \leq s \leq 1$, and to a set of variables $\boldsymbol{\chi}_{f_{\text{traj}}}$ that enable reactive adaptation of the trajectory. These DOF are embedded as feature variables in the control problem in section V. The trajectory is defined by a position profile $\boldsymbol{f}_p(s, \boldsymbol{\chi}_{f_{\text{traj}}})$ that encodes information generalized from demonstrations, and an orientation profile $\boldsymbol{f}_\psi(s)$ which is modeled.

A. Position profile

Position paths are generated and reactively adapted within an allowable space encoded from demonstrations, thereby increasing the predictability of our system. We increase flexibility by defining a set of compact basis functions that enable local deformations in the neighborhood of trajectory segments, thus endowing the system with capabilities to

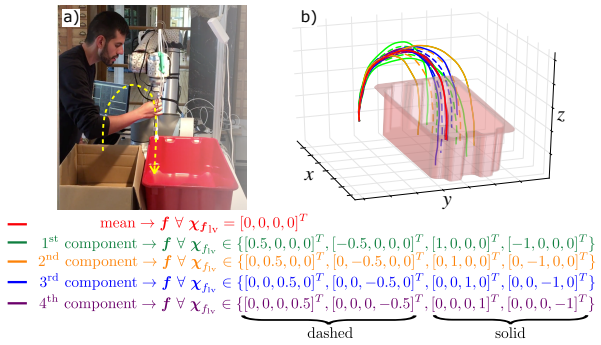

Fig. 2. Learned model to approach to the bin: a) an operator performs a set of demonstrations to encode the desired position path shape, in this case, entering the bin with a vertical approach; b) graphical representation of the variability encoded in each of the basis functions of the motion model in (1).

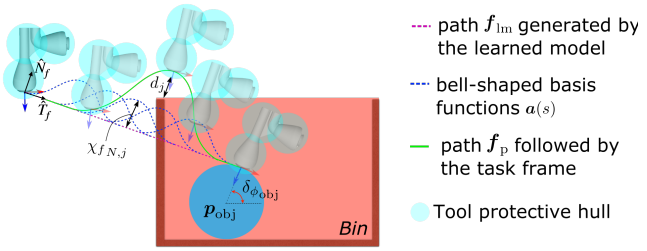

Fig. 3. 2D representation of local trajectory deformations. A task frame located in the tool $\Gamma_{tf}(q)$ moves along f_p to reach the surface of an object located at p_{obj}. Collision between the tool and a bin are avoided by deforming a local section of the path f_{lm} using a piecewise function defined by a set of bell-shaped functions $a(s)$. The amplitudes of the basis functions $\chi_{f_{N,j}}$ are computed by constraining distances d_j, measured between the environment and tool protective hulls distributed along f_{lm}, to be larger than a threshold.

circumvent dynamic obstacles not considered in the demonstrations.

The PPCA algorithm is used to formulate a model defined by a linear combination of basis functions learned from N human demonstrations. This model enables generation of trajectories and their reactive adaptation in a predictable way. As described in [1] and [22], it describes a linear space corresponding to the following probabilistic latent model for the position profiles with $D = 3$ signals and their variability:

$$f_{lm}(s, \chi_{f_{lv}}) = W(s)\chi_{f_{lv}} + b(s), \quad (1)$$

where $W(s) \in \mathbb{R}^{D\times M}$ contains M basis functions extracted from the set of demonstrations, $b(s) \in \mathbb{R}^D$ corresponds to the mean trajectory of the demonstrations, and $\chi_{f_{lv}} \in \mathbb{R}^M$ contains the latent variables of the model, which are directly embedded as feature variables in the control problem in section V. A graphical representation of the probability distribution of a learned motion model is depicted in Fig 2. The figure shows that in this model a larger variability is present along the x-axis, followed by the y-axis, and finally the z-axis.

Local deformations from $f_{lm}(s, \chi_{f_{lv}})$ are enabled using a piecewise function defined along the trajectory with H bell-shaped ($h = 1, ..., H$) basis functions $a(s)$. To this end, these basis functions are defined as continuously differentiable and compact functions, i.e., each individual basis function only induces deformations in its neighborhood, having no influence outside. These functions can be defined using truncated Gaussians, b-splines or any polynomial that satisfies the mentioned properties. As an example, in this paper we choose polynomials of order 4, parameterized by s-interval widths $w_h = \{w_1, \cdots, w_H\}$ and s-interval centers $c_h = \{c_1, \ldots, c_H\}$; then, we project the polynomials in the normal $\hat{N}_f$ and the bi-normal $\hat{B}_f$ direction of $f_{lm}(s, \chi_{f_{lv}})$ modulated by their corresponding amplitude $\chi_{f_N} = \left[\chi_{f_{N,1}} \cdots \chi_{f_{N,H}}\right]^T$ and $\chi_{f_B} = \left[\chi_{f_{B,1}} \cdots \chi_{f_{B,H}}\right]^T$ as follows:

$$f_{bf}(s, \chi_{f_N}, \chi_{f_B}) = \sum_{h=1}^{H} a(s)(\chi_{f_{N,h}}\hat{N}_f + \chi_{f_{B,h}}\hat{B}_f) \quad (2)$$

$$a(s) = \begin{cases} 16s_h^4 - 32s_h^3 + 16s_h^2 & \text{if } c_h - \frac{w_h}{2} \le s \le c_h + \frac{w_h}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\text{with } s_h = \frac{s - (c_h - \frac{w_h}{2})}{w_h}. \quad (4)$$

The amplitudes χ_{f_N} and χ_{f_B} are DOF embedded as feature variables in the control problem in section V.

To find the tangential $\hat{T}_f$, normal $\hat{N}_f$ and bi-normal $\hat{B}_f$ direction, instead of using of the Frenet-Serret formulas in which the normal component is not well-defined for straight lines, these directions are defined as follows:

$$\hat{T}_f = \frac{\frac{df_{lm}}{ds}}{||\frac{df_{lm}}{ds}||}, \quad \hat{N}_f = \frac{\hat{d}_{aux} \times \hat{T}_f}{||\hat{d}_{aux} \times \hat{T}_f||}, \quad \hat{B}_f = \hat{T}_f \times \hat{N}_f, \quad (5)$$

where the auxiliary direction $\hat{d}_{aux}$ is chosen such that $\hat{N}_f$ is more likely to be well defined along the entire trajectory, *i.e.*, $\hat{d}_{aux}$ corresponds to the direction less likely to be parallel to the generated trajectory tangent. Thereby, this direction is chosen by decomposing the tangent vector space of the demonstrations using PCA, ordering the principal values in a descending way, and then, choosing the eigenvector corresponding to the last principal value.

Finally, the position profile $f_p(s, \chi_{f_{traj}})$ is assembled as follows:

$$f_p(s, \chi_{f_{traj}}) = f_{lm}(s, \chi_{f_{lv}}) + f_{bf}(s, \chi_{f_N}, \chi_{f_B}), \quad (6)$$

where $\chi_{f_{traj}} = \left[\chi_{f_{lv}}^T \quad \chi_{f_N}^T \quad \chi_{f_B}^T\right]^T$. Fig. 3 depicts a graphical representation of this reactive trajectory.

B. Orientation profile

To reduce the number of demonstrations, as in [1], a reactively adaptable orientation profile $f_\psi(s)$ is modeled as a linear interpolation using the axis-angle representation in function of s. This profile spans from a known initial orientation at the beginning of the profile $f_\psi(s)|_{s=0}$ towards a variable target orientation at the end of the profile $f_\psi(s)|_{s=1}$.

IV. Allowable Manifolds

The goal pose $\Gamma_G(\chi_{f_{am}}) = \{p_G, R_G\}$ of the generated trajectory is defined within an allowable manifold (hence subscript *am*), specified in function of a set of DOF in which a tool is allowed to move with respect to an object. These DOF are embedded in the control problem in section V and

can be separated into a subset for the surface and another one for the tool ($\boldsymbol{\chi}_{f\,\mathrm{am}} = [\boldsymbol{\chi}_{f\,\mathrm{surf}}{}^T, \boldsymbol{\chi}_{f\,\mathrm{tool}}{}^T]^T$). This separation will be exploited to allow generalization to any combination of surface and tool.

A. DOF related to the surface

The surface of an object, or a section of it, can be modeled with a continuously differentiable function $\boldsymbol{g}(\mathrm{u},\mathrm{v}) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ expressed in frame $\{\mathrm{obj}\}$, where u and v are the generalized coordinates related to the chosen parametrization. In order to determine the goal pose, we first define an intermediate frame whose origin is located at the surface $\boldsymbol{g}(\mathrm{u},\mathrm{v})$ and one of its axes is always aligned with the normal direction (see Fig. 4).

In order to give the controller the freedom to adapt the origin of this frame within the surface, the generalized coordinates are embedded as feature variables in the control problem described in section V, i.e. $\boldsymbol{\chi}_{f\,\mathrm{surf}} = [\mathrm{u},\mathrm{v}]$. Thus, an orientation ${}^{\mathrm{obj}}\boldsymbol{R}_{\perp}(\mathrm{u},\mathrm{v}) = [\hat{\boldsymbol{x}}_{\perp}\ \hat{\boldsymbol{y}}_{\perp}\ \hat{\boldsymbol{z}}_{\perp}]$ (superscript indicating that the axes are expressed in $\{\mathrm{obj}\}$) whose axis $\hat{\boldsymbol{z}}_{\perp}$ remains always normal to the surface must be defined. This axis can be found with (7) and the remaining axes can be chosen arbitrarily. A good option is to choose the axes with (8), where ${}^{\mathrm{obj}}\boldsymbol{R_0} = [\hat{\boldsymbol{x}}_0\ \hat{\boldsymbol{y}}_0\ \hat{\boldsymbol{z}}_0]$ is the initial orientation of the task frame pose, i.e. $\boldsymbol{\Gamma}|_{t=0}$ (see Fig 4).

$$\hat{\boldsymbol{z}}_{\perp} = -\frac{{}^{d\boldsymbol{g}}/_{d\mathrm{u}} \times {}^{d\boldsymbol{g}}/_{d\mathrm{v}}}{||{}^{d\boldsymbol{g}}/_{d\mathrm{u}} \times {}^{d\boldsymbol{g}}/_{d\mathrm{v}}||}, \tag{7}$$

$$\hat{\boldsymbol{x}}_{\perp} = [\hat{\boldsymbol{x}}_0 - (\hat{\boldsymbol{x}}_0 \cdot \hat{\boldsymbol{z}}_{\perp})\hat{\boldsymbol{z}}_{\perp}] \text{ and } \hat{\boldsymbol{y}}_{\perp} = \hat{\boldsymbol{z}}_{\perp} \times \hat{\boldsymbol{x}}_{\perp} \tag{8}$$

B. DOF related to the tool

Additional DOF can be added according to the properties of the tool and to the specific application. An example, illustrated in Fig. 5, shows how additional translational and rotational DOF are defined in some of the axes of the task frame $\{\mathrm{tf}\}$ (i.e. $\hat{\boldsymbol{x}}_{\mathrm{tf}}$, $\hat{\boldsymbol{y}}_{\mathrm{tf}}$ and $\hat{\boldsymbol{z}}_{\mathrm{tf}}$) for a 2-finger gripper and a suction cup. Based on these DOF we create position and rotation sets, $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ respectively, whose elements are function of the DOF (see Fig. 5). The orientation sets are composed by elementary rotation matrices $\hat{\boldsymbol{R}}$ that perform a rotation around one of the axes of $\{\mathrm{tf}\}$ according to the related DOF. The vector that contains all the DOF related to the tool is referred to as $\boldsymbol{\chi}_{f\,\mathrm{tool}}$.

C. Allowable manifold expression

We then find the expression of the allowable manifold in terms of a position (9) and an orientation (10)

$$\boldsymbol{p}_G(\boldsymbol{\chi}_{f\,\mathrm{am}}) = \boldsymbol{p}_{\mathrm{obj}} + \boldsymbol{R}_{\mathrm{obj}}\,\boldsymbol{g}(\mathrm{u},\mathrm{v}) + \sum_{\boldsymbol{\varphi}\in\boldsymbol{\Phi}} \boldsymbol{\varphi} \tag{9}$$

$$\boldsymbol{R}_G(\boldsymbol{\chi}_{f\,\mathrm{am}}) = \boldsymbol{R}_{\mathrm{obj}}\,{}^{\mathrm{obj}}\boldsymbol{R}_{\perp}(\prod_{\hat{\boldsymbol{R}}\in\boldsymbol{\Theta}} \hat{\boldsymbol{R}}) \tag{10}$$

where $\boldsymbol{p}_{\mathrm{obj}}$ and $\boldsymbol{R}_{\mathrm{obj}}$ are the position and orientation of the object, expressed in the world frame. In other words, the goal position (9) is found by transforming $\boldsymbol{g}(\mathrm{u},\mathrm{v})$ to the world frame (first two terms) and then giving it translational freedom in some of the axes of the tool (last term). On the other hand, the goal orientation (10) is found by multiplying the elements of the rotation set, which gives rotational freedom, and then transforming it to the world frame.

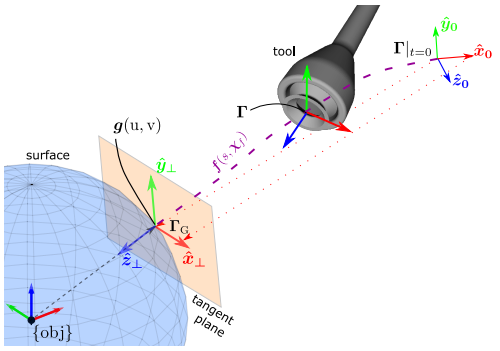

Fig. 4. Graphical representation of a tool approaching towards a surface that models the geometry of an object. Two DOF of the allowable manifold are associated to the surface while additional DOF are associated to symmetries in the tool. The task frame pose $\boldsymbol{\Gamma}(\boldsymbol{q})$ is commanded to evolve along a reactive trajectory $\boldsymbol{f}(s,\boldsymbol{\chi}_f)$ from an initial pose of the task frame $\boldsymbol{\Gamma}(\boldsymbol{q})|_{t=0}$ towards a goal pose $\boldsymbol{\Gamma}_{\mathrm{G}}(\boldsymbol{\chi}_{f\,\mathrm{am}})$. The additional DOF related to the tool are not depicted for ease of understanding.

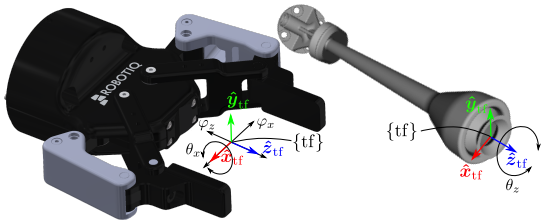

Fig. 5. Example of possible allowable DOF for selected tools, which are depicted with black arrows. $\boldsymbol{\chi}_{f\,\mathrm{tool}} = [\varphi_x\ \varphi_z\ \theta_x]$ for the gripper and $\boldsymbol{\chi}_{f\,\mathrm{tool}} = [\theta_z]$ for the suction cup. The task frame $\{\mathrm{tf}\}$ should be chosen depending on the application, since it will be constrained to reach the allowable manifold at the end of the trajectory (i.e. it can be located with some variation along the $\hat{\boldsymbol{z}}_{\mathrm{tf}}$ direction).

In order to completely define the allowable manifold, inequality constraints are then specified to bound each DOF involved in the definition of the manifold. Some of the bounds depend on the shape properties of both object and tool (e.g. φ_x should be bounded according to the maximum aperture of the gripper and the size of the object). Some others (e.g., u and v) depend on the dimensions of the object. For instance, for a cylindrical surface expressed in cylindrical coordinates, we should bound the coordinate corresponding to the height of the cylinder. It is important to notice that the order of the multiplication in (10) affects the bounding values for orientation, and thus it is recommended to choose the task frame alongside the corresponding Euler angle convention, conveniently.

V. Specification of the Reactive Control

In this section, we highlight the most relevant aspects from *eTaSL* [2], and subsequently introduce the control laws that enable the specification of RAMo. However, it is worth to mention that RAMo is not dependent on eTaSL framework per se, but on its implementation of feature variables that allow to define task-related DOF (i.e. other constraint-based frameworks could also be used instead).

The robot control is expressed by means of eTaSL soft-constraints, which are defined as a function of *joint position variables* $\boldsymbol{q}$, *time variable* t, and *feature variables* $\boldsymbol{\chi}_f$. The latter allow the specification of DOF associated with the task, which can be monitored, controlled, and/or bounded. *eTaSL* formulates the robot behavior as an optimization problem as follows:

$$\underset{\boldsymbol{x}}{\text{minimize}} \qquad \boldsymbol{x}^T \boldsymbol{H} \boldsymbol{x} \tag{11a}$$

$$\text{subject to} \qquad \boldsymbol{L_A} \leq \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{U_A} \tag{11b}$$

where the argument $\boldsymbol{x}$ is a vector $\begin{bmatrix}\dot{\boldsymbol{q}}^T & \dot{\boldsymbol{\chi}}_f^T & \boldsymbol{\varepsilon}^T\end{bmatrix}^T$ that contains: the time derivative of the robot joints $\dot{\boldsymbol{q}}$, the time derivative of the feature variables $\dot{\boldsymbol{\chi}}_f$, and slack variables $\boldsymbol{\varepsilon}$ for each soft-constraint. The *weights* w in the diagonal of $\boldsymbol{H}$ enable the system to deal with conflicting constraints. $\boldsymbol{L_A}$ and $\boldsymbol{U_A}$ are lower and upper bounds of the constraints described by matrix $\boldsymbol{A}$, respectively. As explained in [1], constraints in (11b) can be formulated using a velocity-resolved controller to command a task expression $\boldsymbol{e}(\boldsymbol{q}, \boldsymbol{\chi}_f, t)$ to evolve towards zero with (12), following a first order system with time constant k^{-1}, or to command a task expression $\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\chi}_f, t)$ to follow a desired velocity v with (13). This formulation is defined as follows:

$$\boldsymbol{J}\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\chi}}_f\end{bmatrix} = -k\boldsymbol{e} - \frac{\partial \boldsymbol{e}}{\partial t} + \boldsymbol{\varepsilon}, \tag{12}$$

$$\boldsymbol{J}\begin{bmatrix}\dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\chi}}_f\end{bmatrix} = \boldsymbol{v} - \frac{\partial \boldsymbol{g}}{\partial t} + \boldsymbol{\varepsilon}, \tag{13}$$

where the task function Jacobians $\boldsymbol{J}$ map the time derivatives of states $\boldsymbol{q}$ and $\boldsymbol{\chi}_f$ into the time derivatives of the task variables $\boldsymbol{e}$. Besides equality constraints, *eTaSL* also enables the specification of inequality constraints in (12) and (13).

The formulation of the optimization problem (11a) and (11b) allows the specification and composition of the following robot behaviors. The number of feature variables involved in (11a) is given by the cardinality of the sets defined in sections III and IV, i.e. $\left|\left\{s, \boldsymbol{\chi}_{f_{\text{lm}}}, \boldsymbol{\chi}_{f_{\text{N}}}, \boldsymbol{\chi}_{f_{\text{B}}}, \boldsymbol{\chi}_{f_{\text{am}}}\right\}\right| = 1 + M + 2H + 2 + |\boldsymbol{\chi}_{f_{\text{tool}}}|$.

A. Constraints Related to the Allowable Manifolds

Constraint 1. *Manifold boundaries:* Inequality constraints are used to specify upper and lower bounds for the desired feature variables in $\boldsymbol{\chi}_{f_{\text{am}}}$. The use of the feature variables enables an intuitive specification of these manifold boundaries. The constraint can be applied with (12) directly on each feature variable or over a mathematical expression $e_i(\boldsymbol{\chi}_{f_{\text{i}}})$, where $\boldsymbol{\chi}_{f_{\text{i}}} \subseteq \boldsymbol{\chi}_{f_{\text{am}}}$. For instance, if the top of a cylinder is desired to be reached, the feature variables related to the equation of the plane must be constrained with the expression

$$e_1 := \text{u}^2 + \text{v}^2 \leq R^2, \tag{14}$$

where R is the radius of the cylinder.

Constraint 2. *Preferable Values:* Additional soft constraints can be added to introduce preferable values $\boldsymbol{V}_{\text{pref}}$ for some of the feature variables involved in the definition of the manifold. These constraints are specified with a low weight to allow deviations in case a possible collision is detected. If specified, the corresponding DoFs will return to this preferable value once the obstacle moves away.

$$\boldsymbol{e}_2 := \boldsymbol{\chi}_{f_{\text{i}}} - \boldsymbol{V}_{\text{pref}}, \quad \boldsymbol{\chi}_{f_{\text{i}}} \subseteq \boldsymbol{\chi}_{f_{\text{am}}} \tag{15}$$

B. Constraints Related to the Trajectory Generation

Constraint 3. *Generation of the nominal trajectory:* A trajectory is generated by commanding its end pose $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})|_{s=1}$ to coincide with $\boldsymbol{\Gamma}_G(\boldsymbol{\chi}_{\text{am}})$ defined within the manifold. This relation is specified using (12) with

$$\boldsymbol{e}_3 := \boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})|_{s=1} - \boldsymbol{p}_G(\boldsymbol{\chi}_{\text{am}}) \tag{16}$$

This results in the generation of trajectories towards $\boldsymbol{\Gamma}_G(\boldsymbol{\chi}_{\text{am}})$ that can reactively adapt within the combined spaces defined by the demonstrations and the allowable manifold.

Constraint 4. *Deformations along local sections of the trajectory:* In a collision-free motion, the desired behavior corresponds to the tool moving along the path generated by constraint 3, hence, the feature variables $\boldsymbol{\chi}_{f_N}$ and $\boldsymbol{\chi}_{f_B}$ are commanded to be zero with a soft constraint (12).

$$\boldsymbol{e}_4 := \begin{bmatrix}\boldsymbol{\chi}_{f_N}^T & \boldsymbol{\chi}_{f_B}^T\end{bmatrix}^T \tag{17}$$

However, in case an obstacle is perceived along the path, constraint 17 is violated by increasing the amplitudes in $\boldsymbol{\chi}_{f_N}$ and $\boldsymbol{\chi}_{f_B}$ that correspond to the basis functions in the neighborhood of the obstacle (see Fig. 3). As a result, this constraint enables the system to deform $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})$ to circumvent obstacles.

Constraint 5. *Reactive advancement along the trajectory:* A reactive evolution along the generated trajectory is commanded by specifying the path coordinate s as feature variable and constraining its time derivative $\dot{s}$ to follow a trapezoidal velocity profile v_s defined by a maximum velocity $v_{\max}$ and acceleration $v_{\max}$ [1], [4]. This is formulated using a constraint (13) with

$$g_5 := s \quad \text{and} \quad v = v_s \tag{18}$$

This results in a reactive behavior that enables the end effector to hold its position or to be driven backwards along the path by performing the appropriate interaction, e.g. by a moving obstacle.

Constraint 6. *Follow motion signals:* The robot motion is generated by commanding $\boldsymbol{\Gamma}(\boldsymbol{q}) = \{\boldsymbol{p}(\boldsymbol{q}), \boldsymbol{R}(\boldsymbol{q})\}$ to coincide with the current instance of the generated trajectory $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})$. To this end, the generated trajectory is mapped to a position vector $\boldsymbol{p}_f(s, \boldsymbol{\chi}_{f_{\text{traj}}})$ and an orientation matrix $\boldsymbol{R}_f(s, \boldsymbol{\chi}_{f_{\text{traj}}})$, and a constraint (12) is applied with:

$$\boldsymbol{e}_{6,p} := \boldsymbol{p}(\boldsymbol{q}) - \boldsymbol{p}_f(s, \boldsymbol{\chi}_{f_{\text{traj}}}) \tag{19}$$

$$\boldsymbol{e}_{6,R} := \boldsymbol{\delta}(\boldsymbol{q}, s, \boldsymbol{\chi}_{f_{\text{traj}}}) \tag{20}$$

where $\boldsymbol{\delta}(\boldsymbol{q}, s, \boldsymbol{\chi}_{f_{\text{traj}}})$ corresponds to the axis-angle representation of $\boldsymbol{R}(\boldsymbol{q})\boldsymbol{R}_f(s, \boldsymbol{\chi}_{f_{\text{traj}}})^{-1}$.

C. Constraints Related to the Environment

Constraint 7. *Point-distance-based collision avoidance:* A collision avoidance behavior is specified by computing distances d_j. As depicted in Fig. 3, this distance is measured from a point cloud representation of the environment towards protective hulls [23] located at the tool (subscript tool)and at H instances of $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\mathrm{traj}}})$ (subscript traj). This computation is expedited by representing the point cloud information with k-d trees, representing the protective hull as a set of spheres, and computing the distance d_j between each sphere and the closest point. Then, d_j is commanded to maintain its value above a threshold d_{th} by defining a constraint (12) with

$$e_{7_{\mathrm{tool}}} := d_j - d_{\mathrm{th}} \geq 0 \tag{21}$$

$$e_{7_{\mathrm{traj}}} := d_j - d_{\mathrm{th}} \geq 0 \tag{22}$$

Therefore, this constraint can act directly upon the motion of $\boldsymbol{\Gamma}(\boldsymbol{q})$ by deviating its progress velocity from the specified velocity v_s, or upon $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\mathrm{traj}}})$ by deforming sections of it even outside of the current neighborhood of $\boldsymbol{\Gamma}(\boldsymbol{q})$. As a result, the generated trajectory can adapt to circumvent obstacles before the tool reaches their neighborhood.

D. Additional constraints

Constraint 8. *Joint limits and proximity-based collision avoidance:* Inequality constraints are used to specify bounds on each robot joint position and velocity. Additionally, similarly as in [4], a collision avoidance behavior is composed by defining reactive velocities as a function of proximity signals sensed by an artificial skin [24].

VI. Validation Use Cases

The proposed framework was tested in one simulated and one real use case that involved grasping objects with different geometries. In each use case it was assumed that the following is known: *(i)* the robot platform and its kinematic model (including the tool), *(ii)* the transformation between the vision system and the robot, and *(iii)* the surface model of the objects to pick.

Five demonstrations towards different approach positions $N = 5$ are performed in both use cases to learn the position path using kinesthetic teaching. Fig. 2 shows that variability in position can be captured using only four basis functions, therefore, we choose $M = 4$. Moreover, five basis functions are chosen to enable local deformations along the trajectory $H = 5$.

During execution, the information about the pose, the dimensions of the target objects (i.e. $\boldsymbol{g}(\mathrm{u}, \mathrm{v})$), and a point cloud representation of the environment were provided by a vision system (they were known for the simulated use case).

A. Grasping of objects with gripper

In this use case the task of the robot consists of grasping a target object with the gripper shown in Fig. 5. Demonstrations were performed in a real setup to encode a motion model that ensures that the tool always approaches linearly along a vector normal to the object surface, such that the gripper does not collide with it. This use case was performed in a simulated environment so that it facilitates the visualization of the trajectory reactively adapting to avoid obstacles. Instead of using a point cloud as in the next use case, we used a virtual spherical obstacle whose position is controlled with a joystick.

Fig. 6 illustrates the response of the generated trajectory when a virtual dynamic obstacle interferes with it. The H instances of $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\mathrm{traj}}})$ are displayed to illustrate how the generated trajectory adapts so that it avoids the obstacle, while the goal pose adapts within the allowable manifold.

Signals related to two trials of the use case are shown in Fig. 7. A trial in which a dynamic obstacle causes deviations in the trajectory (as in Fig. 6) is compared to a baseline trial in which the obstacle stays still and out of range. When at least one of the collision distances (refer to Constraint 7) is below the defined threshold, the control parameters deviate as follows: *(i)* $\dot{s}$ decreases when the protective hull of the tool is about to collide with the obstacle, *(ii)* $\boldsymbol{p}_{\mathrm{tf}}$ deviates from $\boldsymbol{f}_{\mathrm{lm}}(s, \boldsymbol{\chi}_{f_{\mathrm{lm}}})$ due to the local deformations (see (6)), *(iii)* the orientation remains almost constant, but slightly deviates because the target orientation adapts within the allowable manifold, *(iv)* $\boldsymbol{\chi}_{f_N}$ and $\boldsymbol{\chi}_{f_B}$ increase the value of the corresponding basis functions amplitudes, thereby enabling the system to avoid collisions, *(v)* the latent variables $\boldsymbol{\chi}_{f_{\mathrm{lm}}}$ of the first three principal components of the learned motion model slightly deviate due to the change in the target position within the allowable manifold. Notice that $\boldsymbol{\chi}_{f_{\mathrm{lm}}}$, corresponding to the last principal component, has large deviations when the obstacle moves away. However, notice from Fig. 2 that this component has a negligible influence in this particular learned model and thus only three basis functions could have been used instead.

TABLE I. Constraint Parameters.

	Param.	**Value**		**Param.**	**Value**
1	k_1	$2\ \mathrm{s}^{-1}$	5	k_5	0
	w_1	50		w_5	0.5
2	k_2	$1\ \mathrm{s}^{-1}$	6	k_6	$1\ \mathrm{s}^{-1}$
	w_2	0.1		w_6	10
3	k_3	$8\ \mathrm{s}^{-1}$	7_{tool}	$k_{7_{\mathrm{tool}}}$	$1\ \mathrm{s}^{-1}$
	w_3	20		$w_{7_{\mathrm{tool}}}$	50
4	k_4	$2.5\ \mathrm{s}^{-1}$	7_{traj}	$k_{7_{\mathrm{traj}}}$	$8\ \mathrm{s}^{-1}$
	w_4	5		$w_{7_{\mathrm{traj}}}$	10

In Table I, the values of the weights w and control gains k for each i-th constraints (see section V are reported. These values were tuned using intuitive heuristics, based on the relative importance of the constraints and their desired reactiveness. Constraint 2, for example, has a relatively low importance (low weight), since it represents a preferable value that the goal pose will reach whenever there is no obstacle. In contrast, the weights for constraint 1 and 7_{tool} are high, because these constraints are not allowed to be violated. Regarding the tuning of the control gains k_i, constraint 6 and 7_{tool} have low values in order to avoid abrupt movements of

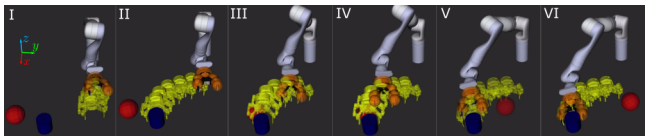

Fig. 6. Snapshots for dynamic obstacle avoidance in simulation. The blue cylinder represents the object to be grasped, the orange spheres depict the protective hull related to the gripper, the yellow grippers represent the location of the H instances of $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})$, and the red sphere is a dynamic obstacle controlled by a joystick.

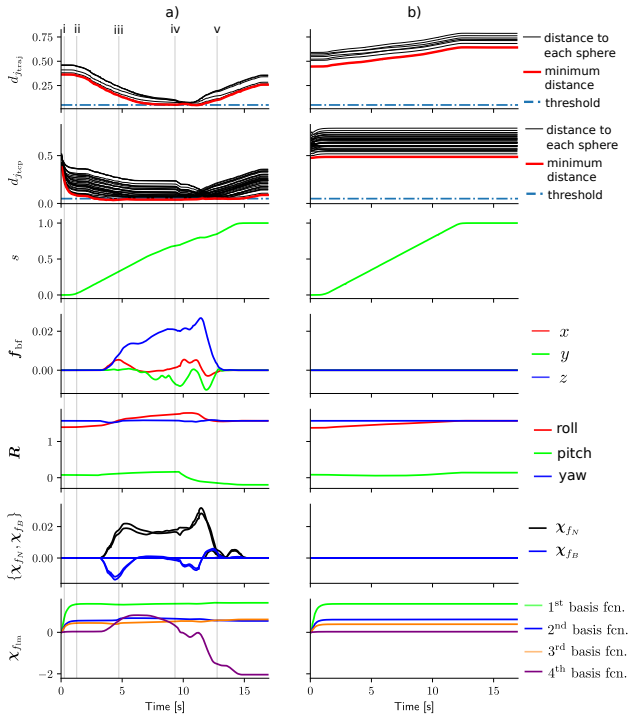

Fig. 7. Control signals related to the grasping of objects with gripper use-case: a) Signals with dynamic obstacle (see Fig. 6) and b) signals without obstacle. The perturbations of the signals, caused due to the obstruction by the obstacle during a part of the trajectory, can be observed. Instances in roman numbers correspond to the snapshots in Fig. 6.

the robot, while the gains related to the trajectory generation are higher to enable faster adaptation.

B. Bin-picking use case

In the second use case, RAMo were used to generate collision-free trajectories that move a tool with a suction cup (see Fig. 5) to pick objects from a cluttered tall bin. Further reactivity is enabled in the system to adapt the generated trajectories in case a human is perceived using proximity sensors as in [4].

The learned motion model in Fig. 2 was demonstrated in such a way that the tool always approaches vertically to the bin, in order to avoid collisions with the walls. Considering the used tool, the total number of feature variables involved in this application was eighteen.

Fig. 1.a shows a tool moving to grasp a sphere in a cluttered environment close to the bin walls. The adaptation of the generated trajectory $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})$ allows the robot to align the suction cup with the spherical surface while avoiding collisions with the environment. The definition of the allowable manifold gives additional flexibility to the

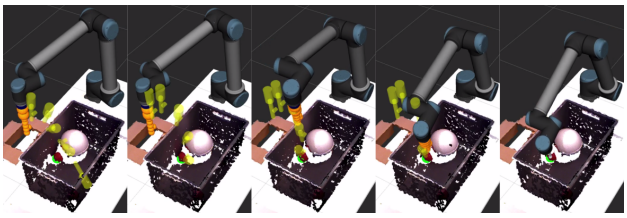

Fig. 8. Snapshots of the point-cloud visualization representing the bin-picking experiment in Fig, 1. A full trajectory generated by the learned model in Fig. 2 is reactively adapted to command its end pose $\boldsymbol{f}(s, \boldsymbol{\chi}_{f_{\text{traj}}})|_{s=1}$ to align with the red sphere. Distances between tool representations (yellow cylinders) and the point-cloud representation of the environment are continuously checked, thereby, enabling the system to avoid collisions.

system by enabling the $\boldsymbol{\Gamma}_G(\boldsymbol{\chi}_{f_{\text{am}}})$ to adapt to avoid collisions (see Fig. 8).

Fig. 1.b shows an operator actively interacting with the robot by approaching his hand towards proximity sensors embedded in an artificial robot skin. Distances between the protective hulls and a point cloud representation of the environment as well as the proximity were continuously checked. During this interaction, the suction cup maintained its alignment with an augmented spherical surface $\boldsymbol{g}(\mathrm{u}, \mathrm{v})$ while avoiding collisions with the environment. The augmented surface was used in this experiments to show the flexibility along the allowable manifold.

VII. Discussion and Conclusion

This paper extends the synergy between LfD and a constraint-based approach by enabling the generation of RAMo towards specified allowable manifolds. In contrast to pure trajectory planners that only rely on the sensed information, the use of LfD enables our system to mitigate sensor limitations that cause poor visibility, e.g. poor representation of the bin walls due to specular surfaces or occlusions.

The achieved reactiveness enables the robot system to reach the goal while avoiding dynamic obstacles and interacting with humans in cluttered environments. The integration of an artificial skin and a vision system allowed us to exploit the aforementioned capabilities in a safe way for the humans.

In contrast to the state of the art, our approach allows a more intuitive definition of the allowable manifolds, which is performed by selecting position and orientation DOF that depend on the object surface and the tool. The two DOF related to the object surface are used to give freedom at the position level, while maintaining the axis of the tool normal to the surface. On the other hand, the geometric properties of the tool can be exploited by the robot task programmer to add more flexibility to the robot skill, at the expense of increasing the number of optimization variables.

The explicit selection of the task variables to learn and the ones to model, expedite the deployment of bin-picking applications. In comparison to pure model-based approaches, this is achieved because the human can easily abstract and demonstrate the context-dependent motion information relevant to achieve the task. On the other hand, in comparison to end-to-end learning methods, this is achieved due to a drastic reduction in the required number of demonstrations.

The orientation component of the approach motion as well as the evolution profile are modeled. These assumptions reduce the variability that need to be learned, therefore, reducing the number of required demonstrations. The number of independent parameters in PPCA grows linearly with the number of learned signals D as opposed to methods based on Gaussian distributions, in which this number increases quadratically (as in [18]). As a result, the *curse of dimensionality* can be mitigated, encouraging further research to extend PPCA to encode orientation, while maintaining the number of demonstrations N relatively small.

Regarding the definition of the allowable manifold, care must be taken since the representation of $\boldsymbol{g}(\mathrm{u}, \mathrm{v})$ could introduce some singularities. In addition, the arbitrary selection of $\hat{\boldsymbol{x}}_{\perp}$ in (8) can induce a singularity if $\hat{\boldsymbol{u}}_{\emptyset}$ is parallel to the normal axis $\hat{\boldsymbol{z}}_{\perp}$.

In case the target objects have a complex geometry, our framework can still be applied by defining the allowable manifold with just DOF related to the tool (provided a goal pose). For example, if a suction cup is mounted, which does not need to be perfectly normal to the surface due to its compliance, infeasible picking poses can be avoided by adapting its orientation.

Future applications of our framework could benefit from a higher level vision-based algorithm which can fit a surface model for the detected object. This could be based on a generic function which can be fitted offline to the point cloud (e.g. a Gaussian Mixture Model), or fitting the most similar shape primitive to it. However, this is out of the scope of our research and we encourage it as future work.

As a conclusion, the proposed method generates RAMo not only preserving information from demonstrations but also reactively adapting to changes in the environment. This reactiveness is achieved by adding freedom in terms of variables related to the task specification. This adds extra flexibility to the system by enabling deviations of the generated trajectory at three different levels: *(i)* deviations in the whole position trajectory that satisfy the information encoded by our LfD method; *(ii)* deviations that are localized at a region of the position trajectory, which extend the flexibility of the learned model; and *(iii)* deviations in the goal pose within an allowable manifold, which is particularly useful for changes in the environment occurring near the object.

In our view, the specification of robot behaviors by monitoring and controlling the task DOF increases the system's predictability and explainability. These aspects are essential to improve the safety and robustness in real robot applications while enabling adaptability to dynamic environments.

REFERENCES

[1] C. A. Vergara, J. De Schutter, and E. Aertbeliën, "Combining imitation learning with constraint-based task specification and control," *IEEE Robot. and Autom. Lett.*, vol. 4, no. 2, pp. 1892–1899, April 2019.

[2] E. Aertbeliën and J. De Schutter, "ETaSL/eTC: A constraint-based task specification language and robot controller using expression graphs," in *IEEE Int. Conf. on Intel. Robots and Systems*, 2014, pp. 1540–1546.

[3] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 61, no. 3, pp. 611–622, 1999.

[4] C. A. Vergara, G. Borghesan, E. Aertbeliën, and J. De Schutter, "Incorporating artificial skin signals in the constraint–based reactive control of human—robot collaborative manipulation tasks," *Industrial Robot*, vol. 46, no. 3, pp. 360–368, 2019.

[5] A. Ambler and R. Popplestone, "Inferring the positions of bodies from specified spatial relationships," *Artificial Intelligence*, vol. 6, no. 2, pp. 157–174, jun 1975.

[6] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[7] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task frame formalism" - A synthesis," *IEEE Trans. Robot.*, 1996.

[8] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, Sep. 2003, pp. 1824–1829.

[9] A. Rodriguez, L. Basaez, and E. Celaya, "A Relational Positioning Methodology for Robot Task Specification and Execution," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 600–611, jun 2008.

[10] N. Somani, M. Rickert, and A. Knoll, "An Exact Solver for Geometric Constraints With Inequalities," *IEEE Robot. and Autom. Lett.*, vol. 2, no. 2, pp. 1148–1155, apr 2017.

[11] N. Somani, M. Rickert, A. Gaschler, C. Cai, A. Perzylo, and A. Knoll, "Task level robot programming using prioritized non-linear inequality constraints," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*. IEEE, oct 2016, pp. 430–437.

[12] G. Borghesan, E. Scioni, A. Kheddar, and H. Bruyninckx, "Introducing geometric constraint expressions into robot constrained motion specification and control," *IEEE Robot. and Autom. Lett.*, vol. 1, no. 2, pp. 1140–1147, July 2016.

[13] R. Krug, T. Stoyanov, V. Tincani, H. Andreasson, R. Mosberger, G. Fantoni, and A. J. Lilienthal, "The next step in robot commissioning: Autonomous picking and palletizing," *IEEE Robot. and Autom. Lett.*, vol. 1, no. 1, pp. 546–553, Jan 2016.

[14] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile Generalized Inverted Kinematics implementation for collaborative working humanoid robots: The Stack Of Tasks," in *International Conference on Advanced Robotics ICAR*, 2009.

[15] È. Pairet, P. Ardón, M. Mistry, and Y. Petillot, "Learning Generalisable Coupling Terms for Obstacle Avoidance via Low-dimensional Geometric Descriptors," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Macau, 2019.

[16] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[17] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Trans. Robot.*, 2011.

[18] M. Ewerton, G. Maeda, D. Koert, Z. Kolev, M. Takahashi, and J. Peters, "Reinforcement Learning of Trajectory Distributions: Applications in Assisted Teleoperation and Motion Planning," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, Macau, 2019, pp. 4294–4300.

[19] A. D. Dragan, N. D. Ratliff, and S. S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.

[20] D. Berenson, S. Srinivasa, and J. Kuffner, "Task Space Regions: A framework for pose-constrained manipulation planning," *International Journal of Robotics Research*, 2011.

[21] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "RMPflow: A Computational Graph for Automatic Motion Policy Generation," 2020.

[22] E. Aertbeliën and J. De Schutter, "Learning a predictive model of human gait for the control of a lower-limb exoskeleton," in *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics*, 2014, pp. 520–525.

[23] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[24] P. Mittendorfer, E. Yoshida, and G. Cheng, "Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot," *Advanced Robotics*, vol. 29, no. 1, pp. 51–67, 2015.